Customized Optimizing McDonald's Meal Application

Jiaming Huang
HEC Lausanne
Bussiness Analytics
jiaming.huang@unil.ch

Manunpat Sirijaturaporn
HEC Lausanne
Bussiness Analytics
manunpat.sirijaturaporn@unil.ch

Ting Yang
HEC Lausanne
Bussiness Analytics
ting.yang@unil.ch

***Abstract*—The fast-food industry has been growing rapidly over the past decades and become a part of the modern lifestyle, especially in urban areas. However, fast food has also been heavily criticized as a major cause of obesity, diabetes, and many diseases around the world. Among the criticisms and research that pointed out the negative health effect caused by fast food, there were some experiments attempting to prove that eating only McDonald's 3 meals a day could help reduce weight. Nevertheless, there are more aspects that one needs to consider, besides weight reduction, to determine the long-term effect on the body, such as the level of cholesterol or sodium. Therefore, this project aims to develop an application that utilizes the nutrition information published by McDonald's and finds the optimal choices for each meal based on individual daily energy intake, estimated using the individual profile. We find that most individuals cannot have adequate calorie intake without over-consuming salt and fat, which proves that items on McDonald's menu generally have a high level of unwanted nutrients such as sodium.**

I. INTRODUCTION

The modern lifestyle is full of hectic schedules, responsibilities, and activities which are related to the times and technology. People, especially urbanite, devote their time to work because of the high cost of living, and the time constraint causes the boom of fast food. Fast food perfectly matches the hustle and bustle since it can be prepared and served within a very short time as a quick meal or to be taken away. It is also usually inexpensive and tastes good. As a result, the popularity of fast food in most parts of the world has been growing at a considerable rate.

With the rise of fast food, however, a healthy lifestyle has become an unstoppable trend due to a bunch of drawbacks of consuming fast food in the long run. Food and nutrients not only play a key role in growth and development but also in the prevention of many diseases. "We are what we eat" is one of the famous food expressions that everyone should agree with. Fast food has come under criticism for being the cause of several illnesses. A poor-quality diet of junk food is linked to various effects on the body ranging from low impacts, such as a high risk of headache and acne breakouts, to high impacts, such as a high risk of obesity, heart disease, stroke, high blood pressure, and depression [2]. Nevertheless, in our point of view, there is some room in the fast-food industry to develop and differentiate.

Speaking of fast food, McDonald's always holds a top rank position due to its long and successful history. McDonald's is the world's leading global food- service retailer with over 38,000 locations in over 100 countries. As the most popular fast-food chain in the world, McDonald's has long been criticized as a cause of illness and disease. A film called 'Fast Food Nation' (2006), written by Linklater and Eric Schlosser, [12] attacked the fast-food industry, especially McDonald's and claimed that McDonald's has been a major cause of making America the fattest nation. A more recent article (Fuhrman, 2018) [5] even stated that eating fast foods may kill more people prematurely than smoking cigarettes. However, a recent experiment (My McDonald's Diet, J. Cisna, 2014) [13] attempted to prove that fast food is not evil by eating McDonald's 3 meals a day for 6 months. In the experiment, the author tried to adhere to a 2,000-calorie diet per day. For example, he chose a bowl of maple oatmeal for breakfast, salad for lunch, and BigMac with fries for dinner. As a result, he managed to lose 60 pounds after 6 months. However, although the experiment in My McDonald's Diet proved that we can deliberately choose dishes from McDonald's menu so that the total calories would not exceed 2,000 calories, it did not prove that McDonald's food is actually "healthy". In our opinion, we also need to consider other nutrients such as Fat, Sodium, and Cholesterol to make sure that they are not over consumed.

II. RESEARCH QUESTION

A. Problem

Nowadays, many fast-food chains publish amounts of calories, Fat, Sodium, and other nutrients on their website, but it is still difficult for customers to understand, and the information is not customized for individuals. For example, on the McDonald's website, McDonald's has information about Sodium per item as a percentage of the daily value, which is the amount that a person should consume each day if the person needs a 2,000-calorie diet. However, the amount of calorie diet that a person need varies from person to person as it depends on several factors, such as height, weight, age, and daily exercise activities. An underweight person might need less than 1,500 calories per day, so the person might over consume Sodium if he or she does not interpret the numbers published by McDonald's carefully.

Hilmers, Angela, et al. (2012) [6] conducted a study in the United States about neighborhood disparities in access to healthy foods and found that low-income neighborhoods have a high density of fast-food outlets and limited access to healthier choices. As a result, this leads to a higher prevalence of obesity in the poor population. Although this paper focuses on the disadvantages of fast food, we find that the strong presence of fast food outlets in poor neighborhoods could benefit the health of people in the area if we could find and promote healthy options from the fast-food menus.

B. Objective

In this research, we would like to conduct a study if we could optimize 3 McDonald's meals a day such that the consumed calories and unhealthy nutrition such as fat, sodium, and sugar don't exceed the level that an individual needs based on the person profiles such as age, gender, weight, height. We will report optimal choices for each meal as well as the calories and other nutrients as a percentage of the amount required by the user.

III. METHODOLOGY

In this part, we would like to illustrate all preparations for the implementation of the design program, including the process of data collection and selection, the creation of various formulas such as optimization problems, and the description of our setup for the program output

A. Data selection

We apply the Nutrition Facts for McDonald's Menu dataset focusing on the United States of America from the Kaggle website [1]. This dataset provides a nutrition breakdown of every menu item on the U.S. McDonald's menu, including breakfast, beef and pork, chicken and fish, fries, salads, snacks and sides, beverages, coffee and tea, smoothies and shakes, and desserts. These data were originally scraped from the McDonald's website before publishing on Kaggle.

B. Optimization problem

a) Basal Metabolic Rate (BMR) and Active Metabolic Rate (AMR): For the first step, we aim to calculate BMR and AMR to determine the number of calories a person needs each day to maintain his or her current weight based on the person's profile, particularly gender, weight, height, age, and sport activity level.

The BMR formula is based on the Harris-Benedict equations [11] as follows:

$$\begin{aligned} BMR_m &= 66.47 + 13.75 \times w + 5.003 \times h - 6.755 \times a \\ BMR_f &= 655.1 + 9.563 \times w + 1.850 \times h - 4.676 \times a \end{aligned} \tag{1}$$

where BMR_m is the BMR for male, BMR_f is the BMR for female, w is weight in kg, h is height in cm, and a is age in years.

And, the AMR can be calculated by multiplying the BMR and the exercise activity as follows:

$$AMR = \begin{cases} BMR \times 1.2 & \text{if Sedentary} \\ BMR \times 1.375 & \text{if Lightly active} \\ BMR \times 1.55 & \text{if Moderately active} \\ BMR \times 1.725 & \text{if Active} \\ BMR \times 1.9 & \text{if Very active} \end{cases} \tag{2}$$

where Sedentary is little or no exercise, Lightly active is exercising 1-2 days/week, Moderately active is exercising 3-5 days/week, Active is exercising 6-7 days/week and Very active is exercising intensively 6-7 days/week.

b) Optimization Problem function: After the AMR is calculated, we perform an integer optimization such that:

$$max \sum_{i=1}^{N} x_i Cal_i \tag{3}$$

Subject to the following constraints:

$$\sum_{i=1}^{N} x_i Cal_i \leqslant AMR \times mealratio$$
$$\sum_{i=1}^{N} x_i TF_i \leqslant 100 \times mealratio$$
$$\sum_{i=1}^{N} x_i SF_i \leqslant 100 \times mealratio$$
$$\sum_{i=1}^{N} x_i Ch_i \leqslant 100 \times mealratio$$
$$\sum_{i=1}^{N} x_i So_i \leqslant 100 \times mealratio$$
$$\sum_{i=1}^{N} x_i Su_i \leqslant 100 \times mealratio$$

where x_i is 1 if item i is selected and 0 if it's not selected, Cal_i is calory for item i, TF_i, SF_i, Ch_i, So_i and Su_i are total fat, saturated fat, cholesterol, sodium and sugar as percentages that the user needs per day for item i, and meal ratio is defined as 35% for breakfast, 35% for lunch and 30% for dinner.

In addition, the percentages of nutrients for each item reported in the input file are calculated based on a 2,000-calorie diet. Thus, we need to adjust for the individual AMR. For example, we can calculate TF_i as follows:

$$TF_i = TF_{original} \frac{AMR}{2000} \tag{4}$$

Furthermore, the amount of sugar for each item reported in the input file is in gram(s). Based on the information from the U.S Department of Agriculture [7] and the World Health Organization [8], 1 gram of sugar can generate 3.87 calories and the total sugar intake should be less than 10% of the total energy intake. We can adjust the constraint for sugar as follows:

$$\sum_{i=1}^{N} x_i Su_{gram} \times 3.87 \leqslant 10\% \times AMR \times mealratio \tag{5}$$

C. Program output

We present the dishes' recommendations on McDonald's menu obtained by optimal calculation to the user through interactive windows. In this program, the user only needs to input their basic personal information, including gender, weight, height, age, and the frequency of exercise activities. Then the program will automatically trigger a series of calculations and data visualization (plotting) functions to present the results to the users on such interactive windows. Thus, apart from the calculation of the optimization problem mentioned above, this program also covers a few other simple calculations and the skills of producing interfaces.

On the other hand, our program is created with Python coding language. During the process of creating the program, we apply two important packages. The first package is Google OR-Tools, which is an open-source software suite for optimization, tuned for tackling the world's toughest problems in vehicle routing, flows, integer and linear programming, and constraint programming, for solving the problem of setting up optimal calculations. The second package is Tkinter [9], which is the standard Python interface to the Tcl/Tk GUI toolkit; moreover, in most of time, TKinter is used with the combination of ttk boostrap which is mainly dedicated to beatify the interface [10]. The specific implementation process will be explained in detail in the following.

IV. IMPLEMENTATION

Our program is hosted in a Github repository, and it contains all the required files and instructions on a step-by-step basis. In the README file, you can find the required installation of the environment and packages which can help to run all our Applications. The program can be run in Python 2022.1 version and is structured using the following packages:

- ortools version: 9.3.10497.
  (This project hosts operations research tools developed at Google and made available as open-source under the Apache 2.0 License.)
- matplotlib version: 3.5.2
- numpy version: 1.22.3
- ttkbootstrap version: 1.7.6
- pandas version: 1.4.2
- Tkinter version: 8.6
- PIL version: 9.1.0

A. Files' structure

To clearly understand the structure of our files, we run the below function to display the directory of all relevant files used in this application.

```
import os
import os.path

def dfs_showdir(path, depth):
   if depth == 0:
      print("root:[" + path + "]")

   for item in os.listdir(path):
      if '.git' not in item:
         print("|    " * depth + "+--" + item)

         newitem = path +'/'+ item
         if os.path.isdir(newitem):
            dfs_showdir(newitem, depth +1)

if __name__ == '__main__':
   dfs_showdir('../Application', 0)
```

Then, we get the structure of our files as follows:

```
root:[../AP_McDonalds]
+--README.md
+--logo
|     +--McDonalds_logo.jpg
+--application
|     +--optimization_calculation.py
|     +--interface_3.py
|     +--interface_2.py
|     +--interface_1.py
|     +--nutrition_plot.py
|     +--main.py
|     +--amr_calculation.py
+--data
|     +--menu.csv
```

In the output of the file structure, we can observe that there are three files, "Interface 1", "Interface 2" and "Interface 3", which are the backbones of the project, mainly dedicated to creating windows, passing important parameters, and triggering relevant calculation functions. Meantime, "main.py" file will be the entrance file which will launch the program and involve TK window to invoke the rest of the program.

The remaining files “amr_calculation”, “nutrition_plot” and “optimization_calculation” are the files that calculate and optimize relevant metrics, and they are imported into each interface file to do the required calculation. In this way, it would help to reduce the duplication of coding. The rest files which are not shown in here are created by the pycharm automatically after running the program.

B. Flow

Our program is built up based on this flow chart. Next, we will explain each component in the order of each interactive page.

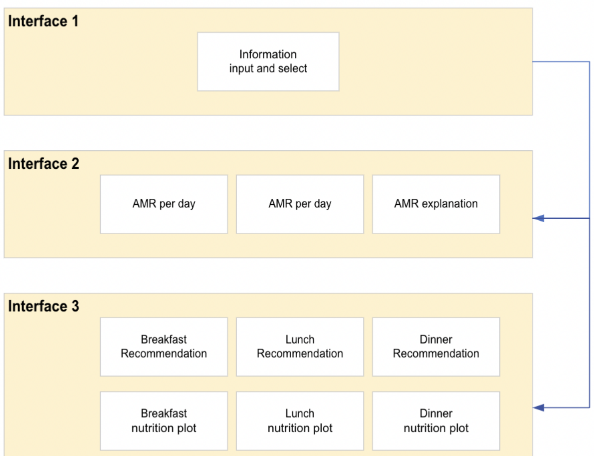

Fig. 1. Interface Composition

From Figure 1, we can observe the main components of each interface. There are three pages in total and they run in order from interface 1 to interface 3.

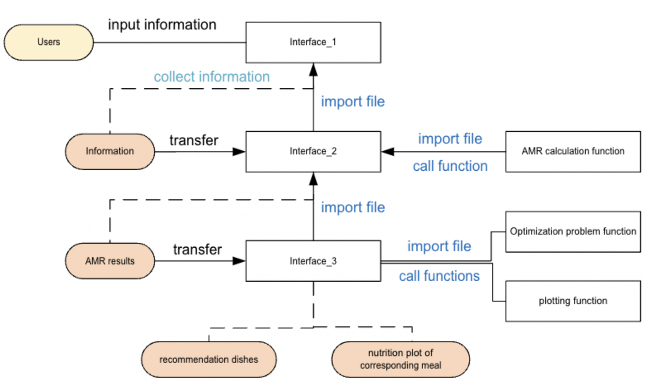

Fig. 2. Information flow and workflow

Then, we can observe the information and workflow of our program in Figure 2. First, the following pages are imported into the previous pages; meantime, some required information in the previous pages will be passed to the next pages. For example, the user information collected in interface 1 will be passed to the class of interface 2. Then, since we need to call the AMR calculation function in interface 2 to present the AMR results, the file of AMR calculation function is imported into interface 2. Finally, interface 3 will display two types of output, recommendation dishes and nutrition plots of the corresponding meal. The details of each file will be explained in-depth below.

a) Main file: The main file which is not shown in the diagram above is for launching the program. We set up this file in order to present a clear structure of the application and keep the main file more independent. It imports the interface_1.py file as the first file to enter. When it triggers the class of interface 1, it passes TK window as the actual parameter to interface 1 which creates the whole window.

b) Interface 1: To build the first page of our program’s interface, we create an individual py.file named “interface_1.py”. The main purpose of this interface is for collecting user information. It contains three main parts: creating a window and organizing the layout of the whole page, users’ information collecting, checking the completeness of information input then jumping to the next page.

- In terms of building up the window and layout organization, there is a 500*550 large window carrying several frames. Then, labels and buttons are inserted into those frames.
- To collect the input of user information, we use the StringVar constructor which helps us manage the value of a widget, and then for these values, we use the get() method of the StringVar object to save as variables used in further steps.
- Since we have to make sure that all information is filled in order to prevent the calculation error due to missing information, we define a function that examines whether all information is complete, then it would pass that information to interface 2.

c) Interface 2:

- The AMR results calculated based on users’ personal information will be presented on this page. In this interface, we calculate the daily AMR and the AMR per meal. Our assumption for the calculation of the AMR for each meal is that 35% of daily calories should be consumed for breakfast and lunch, respectively; and the rest 30% of daily calories is for dinner. The details of the calculation are written in the file named optimization_problem.py file, as mentioned above.
- For the design of interface 2, we first create the window and organize the overall layout as in interface 1. The

purpose of creating two notebooks is to present AMR results and provide an explanation of AMR, to help the user understand the meaning of these values.

- In terms of coding in this part, we import one calculation function, the AMR calculation function from "amr_calculation.py" (equation 1-2). Also, when we call this function we pass the variables such as gender, weight, height, age, and exercise activity frequency as parameters to the function. The AMR calculation function calculates and returns "amr", "breakfast amr", "lunch amr" and "dinner amr" based on the value of the passed parameters and assigns it to "self.amr", "self.breakfastamr", "self.lunchamr", "self.dinneramr". These variables will then be used in the following functions that print the detailed results on the top of this page. On the other hand, the explanation part will be printed at the bottom frame of this page according to the content we already set up.
- Finally, we also set up a button "Next" to pass the calculated AMR value as an argument and move on to the next page. There is in fact a step before moving to the next page on each page which is used to destroy the current page to avoid overlapping contents on both pages.

d) Interface 3:

- Interface 3 receives the "amr", "breakfastamr", "lunchamr" and "dinneramr" variables from interface 2. On this page, we create a notebook format, where each page of the notebook shows our recommendations based on the information obtained from the previous two pages. This means that each page of the notebook is the recommendations for a meal. In addition, at the bottom of the recommendations, we present a graph showing nutrients from the recommended items as a percentage of nutrients that the user should consume per meal based on his or her AMR.
- Therefore, the actual layout of the page differs from the first two pages in that we have added a notebook that carries the three main frames. Within these three frames, we have also added more frames and canvases for the detailed layout and the presentation of the bar charts.
- To obtain the recommendation dishes, we define a function named optimization_calculation. The "breakfastamr", "lunchamr" and "dinneramr" variables are applied to the function to solve the integer optimization problem for each meal using the linear solver wrapper from OR-Tools package. In this program, we include the following 7 steps:
  1) Import the linear solver wrapper.
  2) Declare the LP solver.
  3) Define the variables:
     We limit the maximum number of dishes that can be selected for each item is 1 per meal. If an item is chosen, then the variable is equal to 1, otherwise 0. We also set a limitation that for breakfast only the items in categories of breakfast or beverages can be chosen. And for lunch and dinner, the items in the category of breakfast cannot be chosen. We define the variables by looping over arrays.
  4) Define the constraints:
     We set restrictions on calories and other 5 types of nutrients, which are total fat, saturated fat, cholesterol, sodium, and sugar. We set the first constraint as the total calories provided by all selected items in each meal less than the AMR per meal. We also need to take the consumption of 5 types of nutrients into account such that the user does not over-consume because it is unhealthy to have too much fat, cholesterol, sodium, and sugar. We select the information of the percentage of the daily value of each nutrient provided by each item in the McDonald's menu data set to calculate. However, this percentage is based on 2,000 calories per day so we adjust it according to the AMR we calculated in the previous steps. Regarding sugar, only the amount of sugar in grams is provided and there is no information about the percentage daily value in the data set, so we need to convert the sugar in grams to calories and apply it to build a similar formula. Then we set the constraint as the percentage of the daily value of each nutrient provided by chosen food less than the 100 percent multiplied by meal ratio. We also define the constraints by looping over arrays.
  5) Define the objective:
     The objective function is to calculate the maximum calories for each meal.
  6) Call the LP solver
  7) Display the solution:
     For the output, we would like to display the detailed food recommendations so we provide the food information of id, name, calories, total fat, saturated fat, cholesterol, sodium, and sugar in the final output. Moreover, for each meal, we have food recommendation lists. Therefore, after we run the program it returns the break_output, lunch_output, and dinner_output.

- To obtain the nutrition plot, the meal_plot function receives the arguments of "optimization_output", "mealamr", and "ratio". The "optimization_output" contains the breakfast_output, lunch_output, and dinner_output which are calculated by the optimization_calculation function. The "mealamr" contains "breakfastamr", "lunchamr" and "dinneramr", which are passed from interface 2. The "ratio" is the percentage of daily calories for each meal. The nutrition plot is to show the detailed information of each nutrient per meal by bar plot. Each bar stands for each nutrient, from calories to sugars. Y-axis shows the "dish_values", which are the nutrients provided by recommended dishes as percentages of the total nutrients that are required by each meal. To make the charts more user-friendly, we label the percentage numbers on the top of each bar.

- At the end of the application, we include a button "Quit" for the exit screen, where the command "page.quit" is the page exit command that comes with Tkinter

C. Maintain and update code base

Our project is stored on the github repository, and in the process of writing the project we mainly use pycharm as the writing tool, each time we finish the process of updating the code will go through the action of pull, push, and to link the modified code with the github storage and doing the debugging. In addition, Novolos is also one of our other tools. Also, as the three members of the team use different versions of pycharm and systems, we also try to zip the project folder for passing it to different computers for running and debugging.

V. TEST RESULT

A. Interface running

Here, we use random example to show the test result.

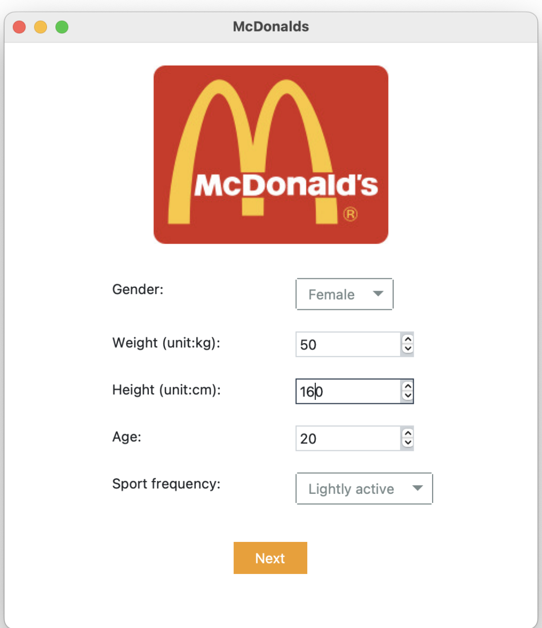

Fig. 3. Interface 1

Figure 3 is the screen of interface 1, on this page, a user can enter their information. Among them, gender and exercise frequency are selected through a drop-down menu, and this step is achieved through menu bottom. weight, height, and age can be entered manually or select a value from arranged values by clicking the arrows, which is achieved by the Spinbox widget. After inputting all information, the user can click "Next" bottom to turn to the next page.

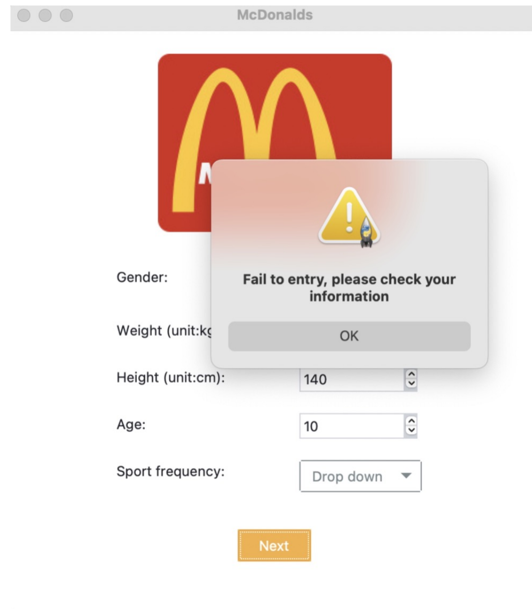

Fig. 4. Interface 1 warning

If the user provides incomplete information, there will be a warning showing as a pop-up window as shown in Figure 4.

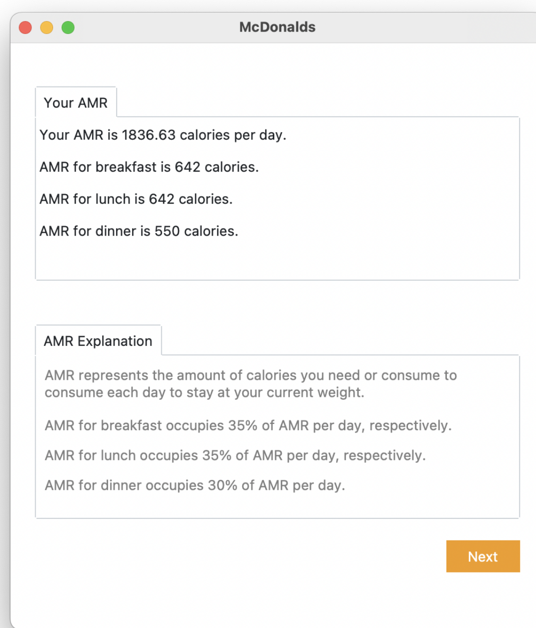

Fig. 5. Interface 2

When it comes to the second page, Figure 5, this page displays the AMR results and the explanation of AMR. Also, the following recommended dishes are provided based on these values. Similar to interface 1, the user can click "Next" bottom to turn to the next page after reading the information. In order to divide the two different areas and display both information clearly at the same time, we set up two notebooks with different font colors to distinguish them.

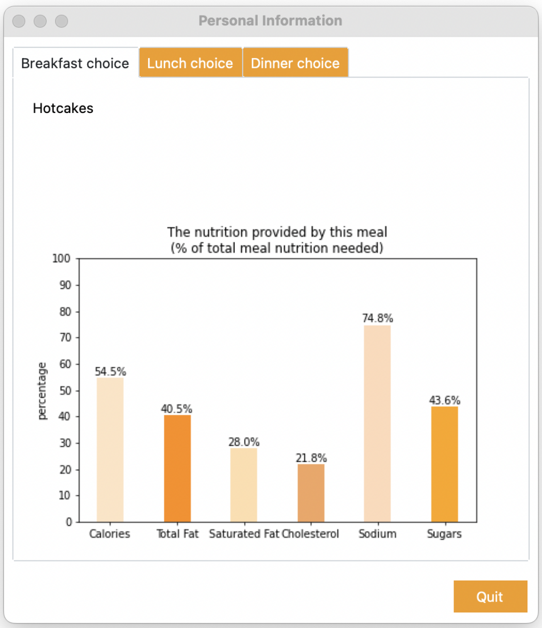

Fig. 6. Interface 3 breakfast page

From the Figures 6 to 8, there are recommended dishes which are the optimal choices calculated by the optimization function on the top of each window. Additionally, there are three pages inside a notebook frame which display the recommendations for each meal separately. Furthermore, at the bottom of each page, there are the nutrition plots of the corresponding meal indicating the percentage of the nutrients that the corresponding meal can provide.

Since there is more information to be displayed on each page, the most difficult part of this page is how to better design the location. Thus, we use more frames nested in this page to assist the layout arrangement. In addition, we also use the canvas component to link the TKinter and matplotlib to present the plotting.

B. Debugging

At the beginning of the program design, we have tried to create an entry file (main.py file) and import each page and the other py files to the main file in order to get a clear and concise programming structure. The purpose is to let the

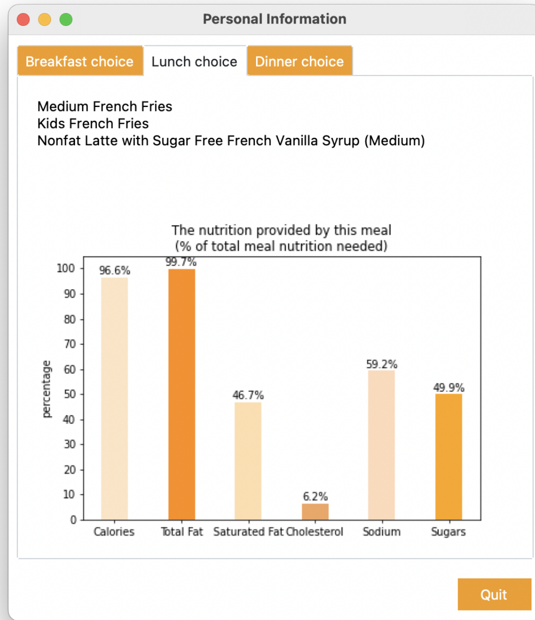

Fig. 7. Interface 3 lunch page

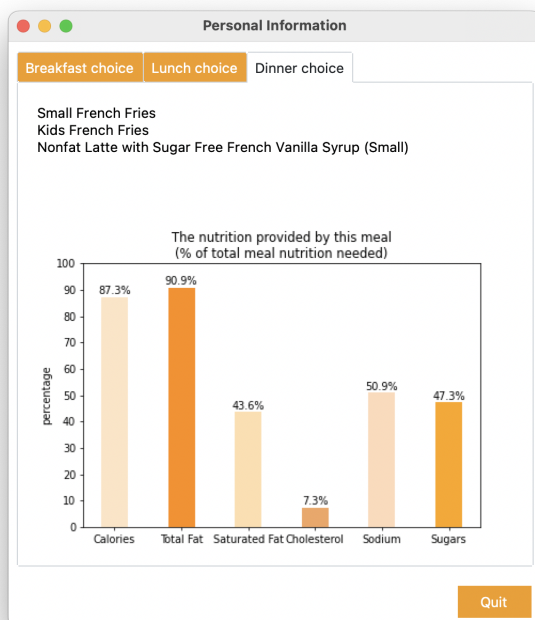

Fig. 8. Interface 3 dinner page

main.py file play the role of contextualization, which means let it start the program, pass parameters and information, call formulas, and so on.

However, we did not find the perfect code flow to do this, because passing the window component of the Tkinter resulted in duplicate passes, which caused the user to have to press the "Next" button twice to jump to another page, which means that the "page.destroy" method does not work properly. Therefore, after several trials, we decided that the most efficient and concise way for us at the moment is as shown in the workflow diagram above.

We also found a problem when calling "tkcanvas" which is one of TKinter methods. When our team member wrote this part, there was a warning message from pycharm that suggested "add a property of this field". However, after the team member followed the instruction to add a property in the script, the script doesn't run properly in the other computers. The revelant pages showed only the recommended content part, but didn't show the charts which were supposed to be created by "tkcanvas". During the debugging process, we re-wrote all the code that has been written step by step to a new file to discover the origin of this problem. To solve this problem, our choice now is not to follow pycharm's suggestions and to simplify the file structure to avoid the possibility of forgetting the file when pushing it because of the complexity of the file structure,

Another problem we encountered in the optimization section was that there was an upper bound of the calorie intake when we set the constraints of five other nutrients (total fat, saturated fat, cholesterol, sodium, and sugar). At first, we did not separate three meals. When we calculated the AMR per day of a user, we applied the total AMR directly as the first constraint to find the maximum calorie intake which would not lead to exceeding the limits of other nutrients. We noticed that when the AMR was larger than 2,000 calories, the maximum calorie intake did not increase. Then we found it was the upper bound of the calorie intake. We separated AMR into three meals in order to solve the problem, then we had AMR per meal. Considering the different foods available at McDonald's at different times, we also limited the list of foods available for each meal, and finally, we had the optimization food recommendations for each meal. It is more suitable and friendly to the user.

C. Possible improvement

As an optimal menu recommendation application, it should be possible to provide users with menu recommendations that meet nutritional needs while meeting the target AMR. However, due to nutrition limitation of McDonald's menu, it is difficult to offer users a combination of perfect recommendation dishes which and satisfy the restriction of calories and requirement of each nutrition intake. Thus, in our application, we should add additional dish recommendations to the existing ones. We have tried to implement this idea, but due to the lack of data it is not possible at the moment. In order to implement this idea, we had the following considerations in mind. Firstly, nutrient information and the units of the new data set for the additional recommendations must be consistent with the information from McDonald's. Secondly, since this application is based on the menu of a certain brand for recommendations, we need to prioritize the selections by choosing items from the menu of the fast food brand first, and then choosing the remaining items from the additional recommendation list. This issue should be achieved by the relevant method of loop nesting. Furthermore, the design of our program can be improved, such as the arrangement of the layout, decoration, etc.

VI. CONCLUSION AND FURTHER DEVELOPMENT

After finishing this McDonald's recommendation application, we let many people with different profiles try to use it. After many attempts, we notice that it is difficult to have adequate calorie intake without exceeding salt and fat intake by only eating McDonald's food, especially for breakfast. This is because the items on McDonald's menu tend to have high sodium and fat. And the choices of food available for breakfast is much less than for lunch and dinner. We are curious about whether other fast food brands will have a similar situation. So for the development of this program, we would like to apply the nutrition facts for the food of other fast food brands such as KFC, Burger King, and Subway, to our program and find which fast-food brand is most likely to provide food that can fulfill calorie needs without exceeding the limit of other nutrition. After we combine all these fast-food brands in the application, the user can choose which fast-food brand he/she wants to eat, and then the program can give food recommendations for each meal of the fast-food brand the user chosen.

Despite the negative sides mentioned above, McDonald's are still the places that provide cheaper and quicker foods, than many healthier options. In addition, McDonald's has a strong presence all around the world. We believe that if McDonald's is able to add more healthier items, especially for breakfast, to provide enough calorie needs without exceeding sodium or fat levels, people around the world will gain benefits by being able to access a cheap, quick and healthy foods.

In addition, such application can be considered in combination with the exercise tracking tool. Since AMR calculations are somewhat subjective, as the choice of exercise frequency is vaguely defined, they do not always accurately reflect the number of calories a person will burn during the day. If it can be used in combination with a system such as a smartwatch for sports detection, the results should be better and it would be more widely used.

REFERENCES

[1] "Nutrition Facts for McDonald's Menu." Www.kaggle.com, www.kaggle.com/datasets/mcdonalds/nutrition-facts. Accessed 30 Mar. 2022.

[2] Ann Pietrangelo. The Effects of Fast Food on the Body [online]. Healthline. September 17, 2018. https://www.healthline.com/health/fast-food-effects-on-body.Accessed 30 Mar. 2022.

[3] "Nutritional challenges and health implications of takeaway and fast food" Academic.oup.com, https://academic.oup.com/nutritionreviews/article/71/5/310/2460221. Accessed 30 Mar. 2022.

[4] "Linear Optimization — OR-Tools." Google Developers, developers.google.com/optimization/lp/lp. Accessed 5 Apr. 2022.

[5] Fuhrman J. The Hidden Dangers of Fast and Processed Food. Am J Lifestyle Med. 2018;12(5):375-381. Published 2018 Apr 3. doi:10.1177/1559827618766483

[6] Hilmers, Angela et al. "Neighborhood disparities in access to healthy foods and their effects on environmental justice." American journal of public health vol. 102,9 (2012): 1644-54. doi:10.2105/AJPH.2012.300865

[7] "FoodData Central." Usda.gov, 2019, fdc.nal.usda.gov. Accessed 5 Apr. 2022. Fast Food Nation My McDonald's Diet - Sugar

[8] World Health Organization. "WHO Guideline : Sugar Consumption Recommendation." Www.who.int, 4 Mar. 2015, www.who.int/news/item/04-03-2015-who-calls-on-countries-to-reduce-sugars-intake-among-adults-and-children. Accessed 5 Apr. 2022.

[9] Python Software Foundation. "Tkinter — Python Interface to Tcl/Tk — Python 3.7.2 Documentation." Python.org, 2019, docs.python.org/3/library/tkinter.html. Accessed 10 Apr. 2022.

[10] "Ttkbootstrap - Ttkbootstrap." Ttkbootstrap.readthedocs.io, ttkbootstrap.readthedocs.io/en/latest/. Accessed 10 Apr. 2022.

[11] "Harris-Benedict -Basal Metabolic Rate Calorie Calc." GlobalRPH, globalrph.com/medcalcs/harris-benedict-basal-metabolic-rate-calorie-calc. Accessed 5 Apr. 2022.

[12] "Fast Food Nation", https://filmsfortheearth.org/en/film/fast-food-nation/. Accessed 1 Apr. 2022.

[13] Cisna, J. "My McDonald's Diet" (2014)